

情報理工学部 SN コース 3 回
リレーショナルデータベースにおけるモデリング

2600200443-6
Yamashita Kyohei
山下 恭平

Jun 19 2022

1 どのような関数従属性が存在しているか

このスキーマには以下の3つの関数従属性が考えられる。

- 顧客 ID → 顧客名, 所在地
- 商品 ID → 商品名, 単価
- 顧客 ID, 商品 ID → 個数, 合計価格

一つ目の「顧客 ID → 顧客名, 所在地」を考えた時、「顧客名 → 顧客 ID, 所在地」も考えられるが、もし同名の顧客が存在した場合に正しく結果を得られることができない。また、「顧客名, 所在地 → 顧客 ID」の場合についても、同名かつ所在地が同じだった場合にエラーが発生する。

二つ目の「商品 ID → 商品名, 単価」を考える。企業のデータベースであるので商品名が一致することはないと考えられる。この時「商品 ID → 商品名 → 単価」の推移的関数従属があるように見えるが、「商品 ID ⇔ 商品名」の関係が存在するので、推移的関数従属はないと考えられる、ここで、商品 ID をキーとする理由は、商品名が変更になった際のエラーを防ぐためである。

三つ目では、一つ目と同様の理由で、個数と商品価格が一致したときにエラーが発生するので ID 二つをキーとする。

2 キーは何か

第一正規系のとき、それぞれの関数従属性よりキーは (顧客 ID, 商品 ID) の二つである。

上の問題で示した、3つそれぞれの関数従属についてテーブルを作成したと考え、K をキー属性、FK を外部キーとすると以下の図のようになる。

顧客
顧客ID (K)
顧客名
所在地

商品
商品ID (K)
商品名
単価

購入情報
顧客ID (K,FK)
商品ID (K,FK)
個数
合計価格

3 どの正規化のレベルを満たしているか

主キーが全てのテーブルで一致しているので、一つにまとめることができる。このときは、第一正規系となる。関数従属に基づき、テーブルを分解した場合は第三正規系となる。

表 1 第一正規系

顧客 ID	顧客名	所在地	商品 ID	商品名	単価	個数	価格合計
A001	大津電子	大津市	1010	3mm ネジ	10	300	3000
A001	大津電子	大津市	1011	丸形プラグ	200	150	30000
A002	草津精工	草津市	1010	3mm ネジ	10	200	2000
A002	草津精工	草津市	1011	丸形プラグ	200	120	24000
A002	草津精工	草津市	1045	5m 銅線	500	50	25000

表 2 第三正規系

顧客 ID	顧客名	所在地	商品 ID	商品名	単価	顧客 ID	商品 ID	個数	価格合計
A001	大津電子	大津市	1010	3mm ネジ	10	A001	1010	300	3000
A002	草津精工	草津市	1011	丸形プラグ	200	A001	1011	150	30000
			1045	5m 銅線	500	A002	1010	200	2000
						A002	1011	120	24000
						A002	1045	50	25000

4 BCNF になるように分解せよ

第三正規系の形の時、「商品 ID \leftrightarrow 商品名」が存在しており、非キー属性から、主キー属性への関数従属が存在したので、これを無くせばボイスコッド正規系となる。以下は、ボイスコッド正規系で分解したものである。ここで、(主) は主キーを、(外) は外部キーを表す。

顧客 ID(主)	顧客名	所在地	顧客 ID(主, 外)	商品 ID(主, 外)	個数	価格合計
A001	大津電子	大津市	A001	1010	300	3000
A002	草津精工	草津市	A001	1011	150	30000
			A002	1010	200	2000
			A002	1011	120	24000
			A002	1045	50	25000

商品 ID(主, 外)	単価	商品 ID(主)	商品名(主)
1010	10	1010	3mm ネジ
1011	200	1011	丸形プラグ
1045	500	1045	5m 銅線

5 BCNF に分解したあとに構築する索引の検討

商品単価で検索範囲を絞りたいので、(商品 ID、単価) を持つテーブルに対して、属性「単価」へ B+ 木構造を持つ索引を付与することで、指定された範囲の単価を持つ、商品 ID を通常より早く探索することができる。

6 「価格合計」の冗長性と、それに伴い発生する不整合についての検討

ここでの価格合計は、あらかじめ価格合計を計算しておき、データベースに格納したものと考えられ、「単価 × 個数」で求められるので、冗長性を保つといえる。しかし、何らかの原因で単価の値が更新されたとき、格納されている値と、計算で得られる値が一致せず、不整合が発生すると考えられる。